Contents

1 Q1:Hadoop Ecosystem Technologies Discussion (10 points)

1. **YARN (Yet Another Resource Negotiator)**: YARN is crucial for resource management in Hadoop. It allocates system resources (like CPU and memory) to various running applications. It improves cluster efficiency and allows for simultaneous data processing, making it vital for managing big data workloads.

2. **Zookeeper**: This is a coordination service for distributed applications. In the context of Hadoop, Zookeeper maintains configuration information, provides distributed synchronization, and manages a cluster's naming registry. It ensures high availability and reliability of the system, crucial for big data applications that can't afford downtime or data inconsistency.

3. **Oozie**: Oozie is Hadoop's scheduler. It's used to manage and schedule Hadoop jobs, such as MapReduce and Pig. Oozie simplifies complex job workflows, allowing you to define a series of jobs to be executed in a specific order. It's essential for automating and optimizing big data processes, ensuring tasks are performed efficiently and on schedule.

4. **Sqoop/Hue**: Sqoop is a tool designed to transfer data between Hadoop and relational databases. It's crucial for integrating big data with traditional data warehouses, allowing for efficient data import/export. Hue, on the other hand, is a web interface for Hadoop services. It simplifies working with Hadoop, providing a user-friendly interface for interacting with various components like HDFS, MapReduce, and YARN. Hue makes it easier to manage big data workflows, especially for those who might not be as comfortable with command-line operations.

2 Q2:Pig Script Commands and Data Processing (10 points)

2.1 Pig Code

```
literate
-- Load daily stock data
daily = LOAD "s3://bigdata-hw6-lanston/pig/NYSE_daily" USING
    PigStorage('\t') AS AS (
    exchange: chararray,
    stock: chararray,
    date: chararray,
    open_price: float,
    high_price: float,
    low_price: float,
    close_price: float,
    volume: long,
    adj_close: float
);

-- Load dividends data
dividends = LOAD "s3://bigdata-hw6-lanston/pig/NYSE_dividends"
    USING PigStorage('\t')  AS (
    exchange: chararray,
    stock: chararray,
    date: chararray,
    dividend: float
);


-- Join the datasets on stock and date
joinedData = JOIN daily BY (stock, date), dividends BY (stock,
    date);

-- Calculate dividend/close_price
calculatedData = FOREACH joinedData GENERATE daily::stock AS
    stock, daily::date AS date,
    dividends::dividend/daily::close_price AS div_close_ratio;

-- Group all
calculatedData2 = GROUP calculatedData ALL;

-- Calculate the min and max ratios
calculatedData3 = FOREACH calculatedData2 GENERATE
    MIN(dividend_ratio.ratio) AS min_ratio,
```

```
    MAX(dividend_ratio.ratio) AS max_ratio;

-- Join to find the records with min and max ratios
min_record = JOIN joinedData BY ratio, min_max_ratios BY
    min_ratio;
max_record = JOIN joinedData BY ratio, min_max_ratios BY
    max_ratio;


-- Prepare the final records for min and max
min_record_final = FOREACH min_record GENERATE
    FLATTEN(dividend_ratio::stock) AS stock,
    FLATTEN(dividend_ratio::date) AS date,
    dividend_ratio::ratio;

max_record_final = FOREACH max_record GENERATE
    FLATTEN(dividend_ratio::stock) AS stock,
    FLATTEN(dividend_ratio::date) AS date,
    dividend_ratio::ratio;

-- Store the final results
STORE min_record_final INTO
    's3://bigdata-hw6-lanston/pig/pig_min' USING PigStorage(',');
STORE max_record_final INTO
    's3://bigdata-hw6-lanston/pig/pig_max' USING PigStorage(',');
```

2.1.1 Q2 SnapShot

Figure 1: [pig shell

3 Q3:Hive Queries (10 points)

3.1 HIVE Code

```
literate
-- Create an external table nyTaxi
CREATE EXTERNAL TABLE nyTaxi (
    VendorID INT,
    lpep_pickup_datetime STRING,
    lpep_dropoff_datetime STRING,
    store_and_fwd_flag STRING,
    RatecodeID INT,
    PULocationID INT,
    DOLocationID INT,
    passenger_count INT,
    trip_distance FLOAT,
    fare_amount FLOAT,
    extra FLOAT,
    mta_tax FLOAT,
    tip_amount FLOAT,
    tolls_amount FLOAT,
    ehail_fee STRING,
    improvement_surcharge FLOAT,
    total_amount FLOAT,
    payment_type INT,
    trip_type INT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
STORED AS TEXTFILE
LOCATION 's3://bigdata-hw6-lanston/hive/';

-- Get distinct RatecodeID from the table
SELECT DISTINCT RatecodeID FROM nyTaxi;

-- Show all rows/columns where RatecodeID = 1
SELECT * FROM nyTaxi WHERE RatecodeID = 1;
```

3.1.1 Q3 SnapShot

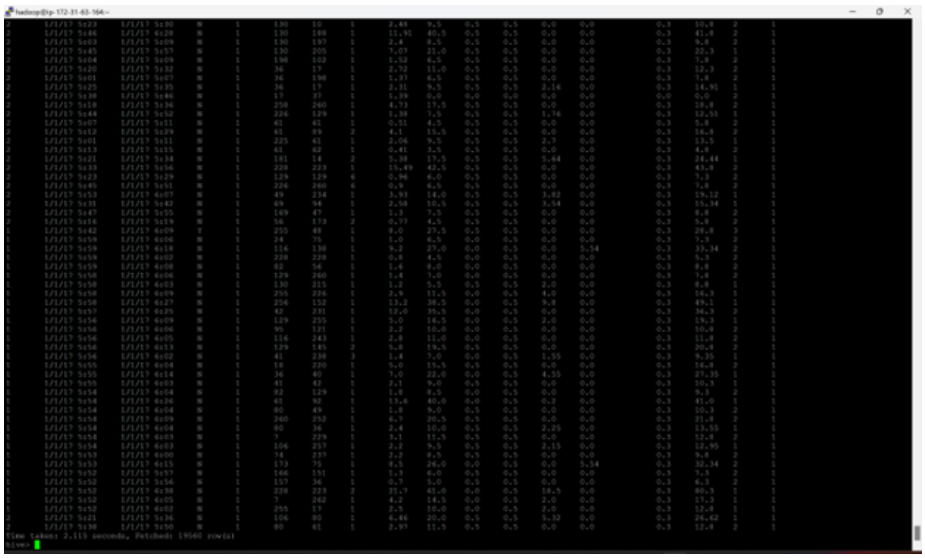

Figure 2: [hive shell

4 Q4:Weblog Data Analysis (10 points)

4.1 Q4 Code

```
literate
CREATE EXTERNAL TABLE  tripadvisor_logs (
    `ip` STRING,
    `timestamp` STRING,
    `request` STRING,
    `status` INT,
    `bytes` BIGINT,
    `referrer` STRING,
    `useragent` STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
STORED AS TEXTFILE
LOCATION 'hdfs:///hive/bigdata/hw6/tripadvisor';


SELECT ip, COUNT(*) AS error_count
FROM tripadvisor_logs
WHERE status = 404
GROUP BY ip;
```

4.1.1 Q4 SnapShot

```
    > GROUP BY ip;
Query ID = hadoop_20231115003150_e6a89465-ddcf-4bab-a0ef-333353789d2d
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1700004796704_0005)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 8.20 s
```

Figure 3: [shell